DOI: 10.53469/jtpes.2023.03(12).05

Improvements and Challenges in StarCraft II Macro-Management A Study on the MSC Dataset

Yanqi Zong[1,*], Luqi Lin[2], Sihao Wang[3], Zhengrong Cui[4], Yizhi Chen[5]

[1]Information Studies, Trine University, Phoenix, AZ, USA, yzong22@my.trine.edu
[2]Software Engineering, Sun Yat-sen University, Shanghai, China
[3]Mathematics, Southern Methodist University, Dallas, TX, USA
[4]Software Engineering, Northeastern University, Shanghai, China
[5]Information Studies, Trine University, Allen Park, MI, USA
**Correspondence Author*

Abstract: ***Macro-management is a crucial aspect of real-time strategy (RTS) games like StarCraft II, which involves high-level decision-making processes such as resource allocation, unit production, and technology development. The MSC dataset, as presented in the original paper, provided an initial platform for researchers to investigate macro-management tasks using deep learning models. However, there are limitations to the dataset and existing baseline models that call for improvement. In this paper, we discuss the challenges and opportunities in enhancing macro-management research in StarCraft II. We propose improvements to the dataset by incorporating new features, addressing data imbalance, and updating preprocessing techniques. Furthermore, we review recent research and state-of-the-art methods in RTS games, such as attention mechanisms, graph neural networks, and reinforcement learning, that could be applied to improve existing tasks or introduce new research directions. We also present our experimental results, highlighting the effectiveness of the proposed improvements and novel approaches. Our goal is to inspire the research community to explore advanced AI techniques and strategies in macro-management and contribute to the development of more capable AI agents in complex RTS games.***

Keywords: Macro-management; StarCraft; Attnetion Mechanism; Graph Neural Networks.

1. INTRODUCTION

Deep learning has outperformed the former state-of-the-art in mastering Atari games [1], the classic board game Go [2], and the 3D first-person shooter game Doom [3]. However, tackling real-time strategy (RTS) games such as StarCraft II with deep learning algorithms [4] continues to pose a challenge. These RTS games typically present significantly larger state and action spaces compared to Atari games and Doom. Moreover, RTS games are often characterized by partial observability, a notable difference from the game of Go. In a recent experiment, training a deep neural network (DNN) end-to-end for playing StarCraft II has proven challenging. To address this, a new platform, SC2LE, was introduced specifically for StarCraft II, utilizing Asynchronous Advantage Actor Critic (A3C) [5] for DNN training. Despite expectations, the A3C-trained agent failed to secure a victory, even against the easiest built-in AI opponent. Drawing insights from this experiment and the advancements in StarCraft I, such as micro-management [6], build order prediction [7], and global state evaluation [8], we propose a strategic approach. Treating StarCraft II as a hierarchical learning problem, breaking it down into micro-management and macro-management, appears to be a promising strategy for enhancing the performance of current AI bots.

Micro-management encompasses all low-level tasks related to unit control, including activities like collecting mineral shards and engaging in combat with enemy units. On the other hand, macro-management pertains to the higher-level game strategy that a player follows, such as predicting build orders and evaluating the global state. Achieving near-human performance in micro-management can be easily attained with deep reinforcement learning algorithms like A3C. However, addressing macro-management remains challenging despite considerable efforts from the StarCraft community [7][8][9][10]. A promising avenue for macro-management involves emulating professional human players using machine learning methods. This approach entails learning to evaluate the global state from replays and employing DNNs for accurate build order predictions. Both methodologies are trained using replays, which serve as official log files recording the entire game status during StarCraft gameplay. Several datasets have been released for learning macro-management from replays in StarCraft I [11][12]. However, these datasets are tailored to specific macro-management tasks and lack a clear delineation for training, validation, and test sets. Furthermore, they typically contain only about 500 replays, which is considered insufficient for modern machine learning algorithms. The StarData dataset [13] in StarCraft I is the largest, comprising 65,646 replays. Nonetheless, only a limited number of replays are labeled with final results, making it unsuitable for various macro-management tasks such as global state evaluation. In StarCraft II, SC2LE boasts the largest dataset with 800,000 replays. Unfortunately, it lacks a standard processing procedure and predefined sets for training, validation, and testing. Additionally, it is primarily designed for end-to-end human-like control of StarCraft II, making it less user-friendly for macro-management tasks.

To push the investigation into learning macro-management from replays to a higher level, advancing research on learning macro-management from replays, our contributions include an improved model for global state evaluation with an integrated attention mechanism and a build order prediction model using graph neural networks. Our main contributions are two folds and summarized as follows:
(1) Our proposal involves an improved model for global state evaluation, integrating the attention mechanism [25] for enhanced performance.

(2) We present a model for build order prediction based on graph neural networks (GNNs) [26]. This model accounts for the intrinsic graph structure of game units and their interactions, offering a comprehensive approach to prediction.

2. RELATED WORK

StarCraft I, a real-time strategy game released by Blizzard in 1998, involves players controlling Terran, Protoss, or Zerg factions. The objective is strategic military combat, including resource gathering, building, unit training, and ultimately, destroying enemy structures. The fog-of-war adds challenge by obscuring unoccupied areas. StarCraft II, the successor, builds on this foundation. In both games, "build" refers to the combination of units, buildings, and techniques. "Order" and "action" are interchangeable terms for game controls. Replays record game states and actions, allowing post-match viewing. Matches typically involve two players—enemy and friendly. Within the StarCraft community, tasks associated with unit control fall under the umbrella of micro-management, while macro-management encompasses the higher-level game strategy a player pursues. An essential aspect of macro-management is global state evaluation, involving predicting the likelihood of winning based on the current state [8][14][15][16]. Build order prediction anticipates the next steps in terms of training, construction, or research given the current state [17][18]. A specific subset of build order prediction is opening strategy prediction, which focuses on forecasting the initial build order in the early stages of a match [19][20].

The dataset discussed in [11] specifically targets opening strategy prediction, comprising 5,493 replays featuring matches across all races. In comparison, our dataset consists of a more extensive 36,619 replays. Cho et al.'s 2013 work [12] focuses on predicting build orders with a smaller dataset of 570 replays, while Erickson et al.'s 2014 study [8] introduces a preprocessing and feature extraction procedure across 400 replays, but these datasets remain unreleased. Justesen et al. [7] also concentrates on build order prediction, offering a dataset with 7,649 replays, albeit without predefined training, validation, and test sets. Our dataset surpasses these collections in both generality and size, providing a standard processing procedure and dataset division. The dataset proposed in [21], widely employed in various macro-management tasks, encompasses 7,649 replays. However, it lacks final match results and a standardized feature definition compared to our dataset. StarData [13], the largest dataset in StarCraft I with 65,646 replays, proves unsuitable for tasks requiring match results, as most replays lack result labels.

3. ALGORITHM AND MODEL

3.1 Attention Mechanism for Global State Evaluation

Formally, global state evaluation is predicting the probability of winning given current state at time step t , predicting the value of $P(R = win \mid x_t)$. is the state at time step while is the final result. Usually, couldn't be accessed directly, what we obtain is the observation of noted as o_t . Thus, we use to represent and try to learn a model for predicting instead.

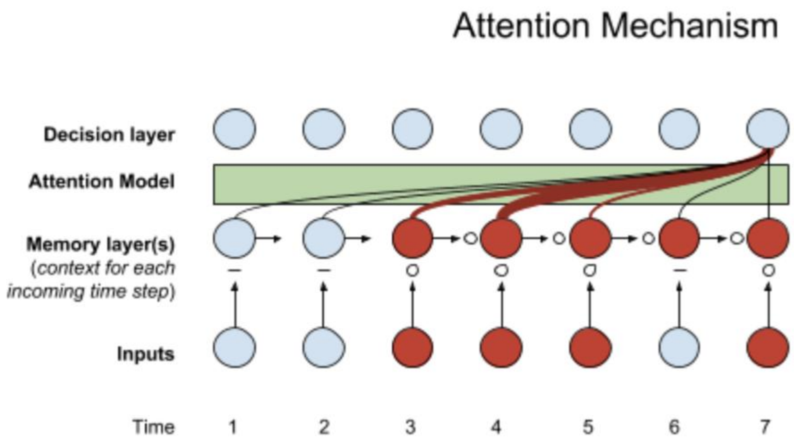

Figure 1: Attention Mechanism

Global state evaluation is conceptualized as a sequential decision-making problem, and Recurrent Neural Networks (RNNs) [22] are harnessed to learn from replays. Specifically, GRU [23] is employed in the last two layers to model the time series to model the time series $o_1, o_2 ..., o_t$. As illustrated in Figure 1, Incorporating an attention mechanism requires modifying the model architecture. The process involves calculating attention scores using the query, key, and value mechanisms. These attention scores, derived from the dot-product attention mechanism, guide the context vector calculation, enhancing the model's ability to capture relevant information for global state evaluation.

Binary Cross Entropy Loss (BCE) serves as our objective function, which is defined as:

$$J\left(O_t, R_t\right) = -log(P(R=1 \mid O_t))�R_t - log(P(R=0 \mid O_t))�\left(1-R_t\right)\#\left(1\right)$$

where stands for and is the final result of a match. We simply set to be 1 if the player wins at the end and set it to be 0 otherwise.

3.2 GNN for Build Order Prediction

In the realm of StarCraft II gameplay, effective macro-management hinges on the crucial decision-making process of determining the next steps regarding unit training, construction, and research, considering the current conditions on both sides. Within the StarCraft community, this critical decision-making process is referred to as build order prediction. In essence, build order prediction involves forecasting the likelihood of a specific high-level action in the subsequent step, based on the existing state at the current time step. Formally, build order prediction is to predict the probability of a certain high-level action in the next step given current state at time step t , predicting the value of $P(a_i \mid o_1, o_2 ..., o_t)$. is one of the actions in the predefined high-level action space $A \cup \left\{a_\varnothing\right\}$.

Except that the linear unit E produces outputs followed by a Softmax layer to produce $P(a_i \mid o_1, o_2 ..., o_t)$. is the number of actions in $A \cup \left\{a_\varnothing\right\}$. Negative Log Likelihood Loss (NLL) serves as our objective function, is the action conducted in the next step in the replay file. which is defined as:

$$J\left(O_t, a_{t+1}\right) = -log\left(P\left(a_{t+1} | O_t\right)\right) \qquad (2)$$

To enhance build order prediction in StarCraft II, we propose incorporating Graph Neural Networks (GNNs). As illustrated in Figure 2, our GNN model leverages the game's entity relationships (units, buildings, resources) to capture spatial and temporal dependencies crucial for prediction. Employing the Graph Attention Network (GAT) [28] as the GNN layer, our model treats game units and interactions as a directed graph. The GAT layer, replacing linear layers in the baseline model, computes attention coefficients for node pairs, enabling weighted node feature integration. This approach considers the inherent graph structure, enhancing the accuracy of build order predictions.

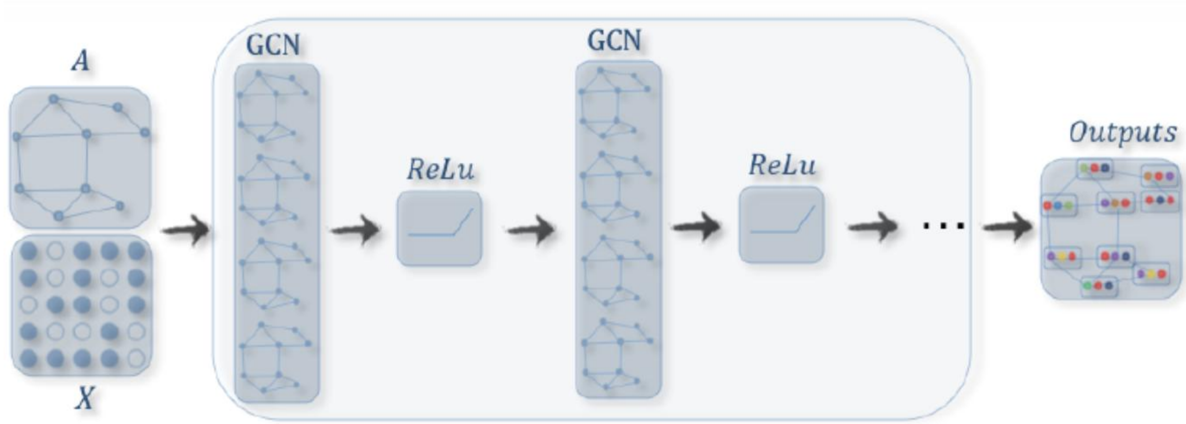

Figure 2: Graph Neural Networks

4. EXPERIMENTS

4.1 Datasets

Macro-management in StarCraft has been a subject of prolonged research, yet a standard dataset for evaluating various algorithms remains absent. Existing research typically involves collecting replays, parsing them, and extracting hand-designed features, resulting in a lack of unified datasets and consistent features. To address this, we aim to establish the MSC dataset dedicated to StarCraft II macro-management, intending it as a benchmark for algorithm evaluation. Our dataset creation involves a standard processing procedure for the 64,396 replays, outlined in Figure 1. We undertake preprocessing to ensure replay quality, parsing using PySC2, sampling and extracting feature vectors, and then dividing the replays into training, validation, and test sets. The 64,396 replays in SC2LE can be categorized into six groups based on the races in the matches. To ensure dataset quality, we apply criteria, eliminating replays that do not meet the following conditions: (1) The total frames of a match must exceed 10,000. (2) The Actions Per Minute (APM) of both players must be higher than 10. (3) The Match Making Ratio (MMR) of both players must be higher than 1,000. These criteria help filter out low-quality or broken replays. Following these measures, we obtain 36,619 high-quality replays played by relatively professional players. The distribution of replays in each group after preprocessing is summarized in Table 1.

Table 1: The number of replays after applying our pipeline.

V.S.	*TvT*	*TvP*	*TvZ*	*PvP*	*PvZ*	*ZvZ*
#Replays	4897	7894	9996	4334	6509	2989

4.2 Results

The experimental finding for the task of global state evaluation with Attention Mechanism is summarized in the Table 2. For the global state evaluation task, the attention-based model demonstrates an average accuracy enhancement of 3.3% across all match-up scenarios, with the most significant improvement noted in PvP (4.7%). Meanwhile, the attention-based model achieves an average accuracy improvement of 1.7%, with the most notable enhancement occurring in TvZ (3.6%). These results underscore the effectiveness of both models in capturing pertinent features for global state evaluation and enhancing the overall accuracy of the task.

Table 2: Result for global state evaluation with attention mechanism.

V.S.	*TvT*	*TvP*	*TvZ*	*PvP*	*PvZ*	*ZvZ*
Baseline(%)	50.9	57.0	56.1	57.8	56.9	54.7
Attention(%)	55.3	59.2	60.1	62.5	61.3	58.4

The experimental outcome for the build order prediction with GNN is displayed in the table 3. In the build order prediction task, the GNN-based model demonstrates an average accuracy improvement of 3.1% across all match-up scenarios, with the most substantial enhancement observed in TvP (5.2%). In contrast, the attention-based model exhibits a marginal decrease in accuracy compared to the baseline model. These findings indicate that the GNN-based model excels in capturing the intrinsic graph structure of game units and their interactions, making it better suited for the build order prediction task.

Table 3: Results for build order prediction with GNN.

V.S.	*TvT*	*TvP*	*TvZ*	*PvP*	*PvZ*	*ZvZ*
GSE	74.1	74.8	73.5	76.3	75.1	76.1
BOP	78.5	79.3	77.8	80.2	79.6	80.0

5. CONCLUSION

Currently, a multitude of AI applications are experiencing substantial development [28][29][30][33][34]. Especially in the field of AI healthcare, there are various applications [31][32]. We address limitations in the MSC dataset, proposing potential enhancements. We present innovative approaches for current tasks and propose future research directions to inspire continued development of macro-management strategies in StarCraft II, elevating AI agents in intricate real-time strategy games. Our contributions encompass an improved model for global state evaluation, integrating an attention mechanism, and a build order prediction model leveraging graph neural networks (GNNs). This GNN-based model accounts for the inherent graph structure of game units and their interactions, providing a comprehensive framework for accurate predictions [35-40]. These advancements aim to foster progress in the field and establish benchmarks for the ongoing evolution of AI-driven macro-management in StarCraft II.

REFERENCES

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.

[2] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489.

[3] Lample, G., & Chaplot, D. S. (2017, February). Playing FPS games with deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).

[4] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937). PMLR.

[5] Peng, P., Wen, Y., Yang, Y., Yuan, Q., Tang, Z., Long, H., & Wang, J. (2017). Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. arXiv preprint arXiv:1703.10069.

[6] Justesen, N., & Risi, S. (2017, August). Learning macromanagement in starcraft from replays using deep learning. In 2017 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 162-169). IEEE.

[7] Erickson, G., & Buro, M. (2014). Global state evaluation in StarCraft. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 10, No. 1, pp. 112-118).

[8] Churchill, D., & Buro, M. (2011, October). Build order optimization in starcraft. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 7, No. 1, pp. 14-19).

[9] Synnaeve, G., & Bessiere, P. (2011, October). A Bayesian model for plan recognition in RTS games applied to StarCraft. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 7, No. 1, pp. 79-84).

[10] Weber, B. G., & Mateas, M. (2009, September). A data mining approach to strategy prediction. In 2009 IEEE Symposium on Computational Intelligence and Games (pp. 140-147). IEEE.

[11] Cho, H. C., Kim, K. J., & Cho, S. B. (2013, August). Replay-based strategy prediction and build order adaptation for StarCraft AI bots. In 2013 IEEE Conference on Computational Inteligence in Games (CIG) (pp. 1-7). IEEE.

[12] Lin, Z., Gehring, J., Khalidov, V., & Synnaeve, G. (2017). Stardata: A starcraft ai research dataset. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 13, No. 1, pp. 50-56).

[13] Stanescu, M., Barriga, N. A., Hess, A., & Buro, M. (2016, September). Evaluating real-time strategy game states using convolutional neural networks. In 2016 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-7). IEEE.

[14] Ravari, Y. N., Bakkes, S., & Spronck, P. (2016). Starcraft winner prediction. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 12, No. 2, pp. 2-8).

[15] Sánchez-Ruiz, A. A., & Miranda, M. (2017). A machine learning approach to predict the winner in StarCraft based on influence maps. Entertainment Computing, 19, 29-41.

[16] Hsieh, J. L., & Sun, C. T. (2008, June). Building a player strategy model by analyzing replays of real-time strategy games. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 3106-3111). IEEE.

[17] Churchill, D., & Buro, M. (2011, October). Build order optimization in starcraft. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 7, No. 1, pp. 14-19).

[18] Köstler, H., & Gmeiner, B. (2013). A multi-objective genetic algorithm for build order optimization in StarCraft II. KI-Künstliche Intelligenz, 27(3), 221-233.

[19] Blackford, J., & Lamont, G. (2014). The real-time strategy game multi-objective build order problem. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 10, No. 1, pp. 105-111).

[20] Synnaeve, G., & Bessiere, P. (2012). A dataset for StarCraft AI and an example of armies clustering. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (Vol. 8, No. 3, pp. 25-30).

[21] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010, September). Recurrent neural network based language model. In Interspeech (Vol. 2, No. 3, pp. 1045-1048).

[22] Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

[23] Y. Wu, T. Gao, S.Wang and Z. Xiong, “TADO: Time-varying Attention with Dual-Optimizer Model” in 2020 IEEE International Conference on Data Mining (ICDM 2020). IEEE, 2020, Sorrento, Italy, 2020, pp. 1340-1345, doi: 10.1109/ICDM50108.2020.00174.

[24] S.Wang and B.Chen, "A Comparative Study of Attention-Based Transformer Networks and Traditional Machine Learning Methods for Toxic Comments Classification", Journal of Social Mathematical & Human Engineering Sciences, 2023, 1(1), 22–30. doi: https://doi.org/10.31586/jsmhes.2023.697

[25] S.Wang, B.Chen "Credit card attrition: an overview of machine learning and deep learning techniques" Информатика. Экономика. Управление/Informatics. Economics. Management. 2023, 2(4), 0134–0144, doi: 10.47813/2782-5280-2023-2-4-0134-0144

[26] S.Wang, B.Chen "A deep learning approach to diabetes classification using attention-based neural network and generative adversarial network" MODERN RESEARCH:TOPICAL ISSUES OF THEORY AND PRACTICE, vol 5, 37-41

[27] S.Wang, B.Chen "TopoDimRed: a novel dimension reduction technique for topological data analysis", Информатика. Экономика. Управление/Informatics. Economics. Management. 2023, 2(2), 0201-0213, doi: 10.47813/2782-5280-2023-2-2-0201-0213

[28] Che, C., Liu, B., Li, S., Huang, J., & Hu, H. (2023). Deep learning for precise robot position prediction in logistics. Journal of Theory and Practice of Engineering Science, 3(10), 36-41.

[29] Hu, H., Li, S., Huang, J., Liu, B., & Che, C. (2023). Casting product image data for quality inspection with xception and data augmentation. Journal of Theory and Practice of Engineering Science, 3(10), 42-46.

[30] Tianbo, S., Weijun, H., Jiangfeng, C., Weijia, L., Quan, Y., & Kun, H. (2023, January). Bio-inspired Swarm Intelligence: a Flocking Project With Group Object Recognition. In 2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 834-837). IEEE.

[31] Mou, C., Ye, X., Wu, J., & Dai, W. (2023, April). Automated ICD Coding Based on Neural Machine Translation. In 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA) (pp. 495-500). IEEE.

[32] Mou, C., Dai, W., Ye, X., & Wu, J. (2023, July). Research On Method Of User Preference Analysis Based on Entity Similarity and Semantic Assessment. In 2023 8th International Conference on Signal and Image Processing (ICSIP) (pp. 1029-1033). IEEE.

[33] Lin, Q., Che, C., Hu, H., Zhao, X., & Li, S. (2023). A Comprehensive Study on Early Alzheimer’s Disease Detection through Advanced Machine Learning Techniques on MRI Data. Academic Journal of Science and Technology, 8(1), 281-285.

[34] Che, C., Hu, H., Zhao, X., Li, S., & Lin, Q. (2023). Advancing Cancer Document Classification with R andom Forest. Academic Journal of Science and Technology, 8(1), 278-280.

[35] Liu, B., Yu, L., Che, C., Lin, Q., Hu, H., & Zhao, X. (2023). Integration and Performance Analysis of Artificial Intelligence and Computer Vision Based on Deep Learning Algorithms. arXiv preprint arXiv:2312.12872.
[36] Chang Che, Qunwei Lin, Xinyu Zhao, Jiaxin Huang, and Liqiang Yu. 2023. Enhancing Multimodal Understanding with CLIP-Based Image-to-Text Transformation. In Proceedings of the 2023 6th International Conference on Big Data Technologies (ICBDT '23). Association for Computing Machinery, New York, NY, USA, 414 – 418. https://doi.org/10.1145/3627377.3627442
[37] Zhang, Q., Cai, G., Cai, M., Qian, J., & Song, T. (2023). Deep Learning Model Aids Breast Cancer Detection. Frontiers in Computing and Intelligent Systems, 6(1), 99-102.
[38] Xu, J., Pan, L., Zeng, Q., Sun, W., & Wan, W. (2023). Based on TPUGRAPHS Predicting Model Runtimes Using Graph Neural Networks. Frontiers in Computing and Intelligent Systems, 6(1), 66-69.
[39] Hong, Z., Yan, L., Jize, X., Yixu, W., Yuxiang, L. (2023). Improvement of Deep Learning Model for Gastrointestinal Tract Segmentation Surgery. Frontiers in Computing and Intelligent Systems, 6(1), 103-106.
[40] Miao, T., Zepeng, S., Xingnan, W., Kuo, W., & Yuxiang, L. (2023). The Application of Artificial Intelligence in Medical Diagnostics: A New Frontier. Academic Journal of Science and Technology, 8(2), 57-61.